# Java Language Conversion Assistant Beta 2

## Introduction

Welcome to the Beta 2 release of the Java Language Conversion Assistant (JLCA). The purpose of the JLCA is to help you move your Java-language projects to C# and the .NET framework. The JLCA converts Visual $J^{++}$ 6.0 projects and Java-language files to C# syntax, and converts code that accesses most JDK level 1.1.4 classes to use the .NET framework directly.

In this document, we discuss the feature set of the JLCA, how to use the tool, and known issues in this release.

## Features

This is the Beta 2 release of the JLCA. This release converts most Java-language constructs, including

- Language: Most statements (if, for, do, while, switch, try, throw, break, continue). Arithmetic, Shift, Bitwise, Logical, Conditional and Relational operators. J Direct directives (@dll.import, @dll.struct, @dll.structmap). Arrays and Local Variable Declarations
- Classes: Class declarations, inner classes and anonymous classes. Hiding, Overriding and Overloading. Interfaces
- Methods: Abstract, Static, Final, Native, Virtual and Synchronized Methods.
- Forms. WFC forms are converted to Windows Forms. Most AWT components are also converted to Windows Forms.
- JDK level 1.1.4 Classes: This release converts about 70% of the Visual $J^{++}$ 6.0 Classes to C# and the .NET framework. These include the following packages:

| | | |
|---|---|---|
| com.ms.dll | com.ms.io | com.ms.lang |
| com.ms.object | com.ms.util | com.ms.wfc.app |
| com.ms.wfc.core | com.ms.wfc.data | com.ms.wfc.data.adodb |
| com.ms.wfc.data.dsl | com.ms.wfc.data.rds | com.ms.wfc.data.ui |
| com.ms.wfc.io | com.ms.wfc.ole32 | com.ms.wfc.ui |
| com.ms.wfc.util | com.ms.win32 | java.awt |
| java.awt.datatransfer | java.awt.event | java.awt.image |
| java.io | java.lang | java.lang.reflect |
| java.math | java.net | java.sql |
| java.text | java.util | |

- If your project contains non JDK level 1.1.4 classes, or classes that are not yet supported, the statements that use them will be copied to C# unchanged.

In addition, this release of the JLCA contains the following online documentation:

- Conceptual topics for how to convert Java-language code to C#,
- Help for the Java Language Conversion Assistant Wizard,
- More than 900 topics that explain what to do with code that isn't automatically converted.

## Improvements over Beta 1

This release contains the following improvements over Beta 1 of the JLCA:

- Approximately a 300% performance improvement for many project types,
- Support for consuming ActiveX controls and COM components,
- Support for converting a directory of Java files, not necessarily in a Visual J$^{++}$ 6.0 project,
- An additional 28,000 JDK level 1.1.4 class and member mappings.

In Addition, support for the following technologies has been added:

- Threads,
- Delegates,
- WFC User Controls,
- Binary resources,
- AWT,
- Anonymous classes,
- Events,
- JavaDocs comments.

## Future Releases

The four most important features we will be adding in future releases are:

- Better support for handling name conflicts, by renaming variables
- More JDK level 1.1.4 class mappings. Future releases will add more support for the following packages:

| | | |
|---|---|---|
| com.ms.awt | com.ms.ActiveX | com.ms.com |
| com.ms.fx | com.ms.ui | com.ms.DirectX |
| com.ms.dxmedia | java.security | java.text.resources |
| com.ms.ui.event | com.ms.ui.resource | com.ms.ui.windowmanager |
| com.ms.util.cab | com.ms.util.ini | com.ms.wfc.ax |
| com.ms.wfc.win32 | | |

- Online help and documentation. Future releases will have more complete documentation for how to convert applications, and what to do for specific classes that cannot be automatically converted to C#.
- Performance. We will be continuing to improve capacity support and robustness in upcoming releases.

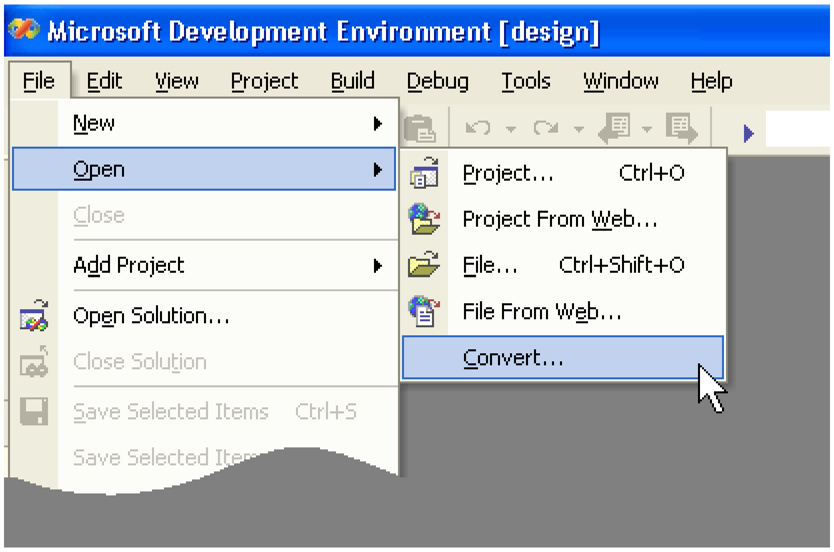
In addition, we are considering adding support for the following:

- JSP page conversion

## Operating Instructions

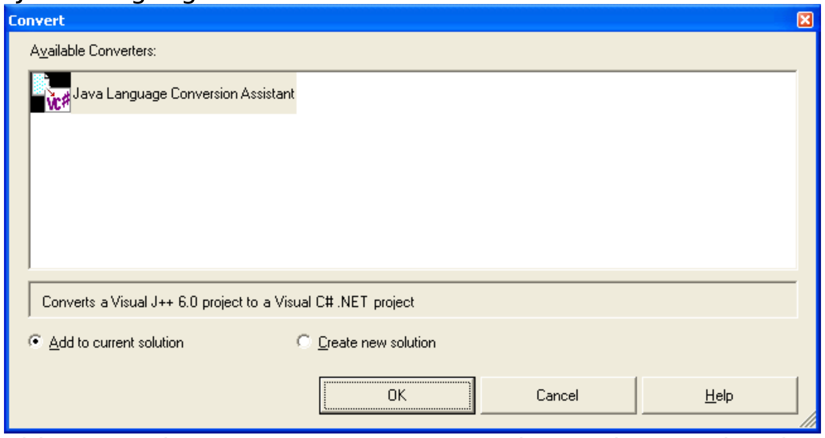
The JLCA wizard is started from the File menu of Visual Studio .NET. Here are step-by-step instructions for how to convert a Java language project to C#:

1. After installing the JLCA, start Visual Studio .NET
2. Choose the File|Open|Convert... menu item.



   This opens the Convert Project Dialog.

3. In the Convert Project Dialog, you select the type of conversion to perform. Select “Java Language Conversion Assistant” and click OK.

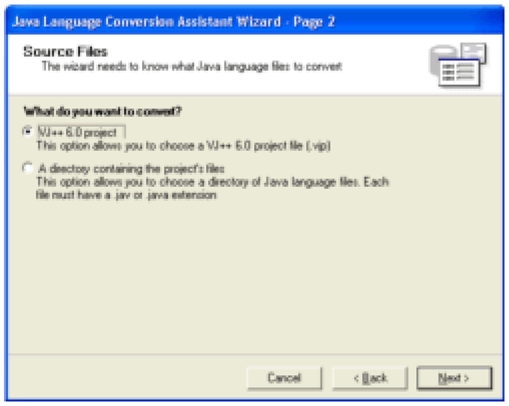


Converts a Visual J++ 6.0 project to a Visual C# .NET project

[✓ Add to current solution] Create new solution

   This starts the Java Language Conversion Assistant Wizard.

4. The wizard has six pages. The first page gives some introductory information about the wizard. The second page allows you to choose whether to convert a VJ$^{++}$ 6.0 project or a directory containing the project’s files.



Java Language Conversion Assistant Wizard - Page 2

Source Files
The wizard needs to know what Java language files to convert

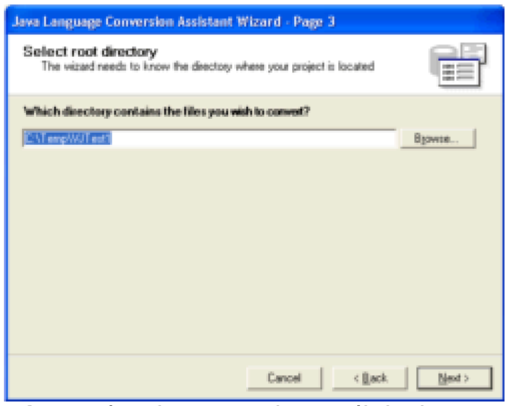
What do you want to convert?

[✓ VJ++ 6.0 project]
This option allows you to choose a VJ++ 6.0 project file (.vjp)

A directory containing the project's files
This option allows you to choose a directory of Java language files. Each file must have a .jav or .java extension

The directory option is useful for Java language projects written in tools other than Visual J++ 6.0.  Click Next to move to Page 3.

5. On Page 3, use the Browse button to select the .vjp project or directory of files to convert



Java Language Conversion Assistant Wizard - Page 3

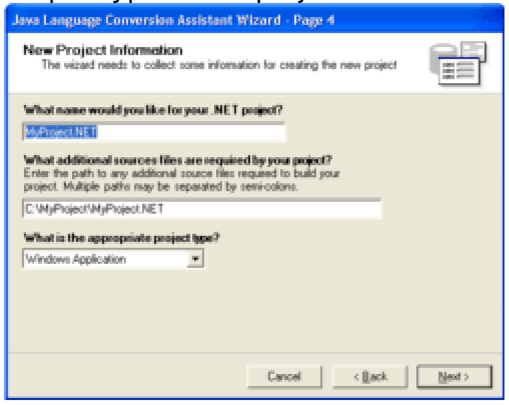
Select root directory
The wizard needs to know the directory where your project is located

Which directory contains the files you wish to convert?

C:\Temp\VJTest1

After selecting a project, click the Next button to move to Page 4.

6. If you selected a directory to convert, Page 4 of the wizard will prompt you to enter name, additional source files location (this can be used to specify the classpath) and output type of the project.



Java Language Conversion Assistant Wizard - Page 4

New Project Information
The wizard needs to collect some information for creating the new project

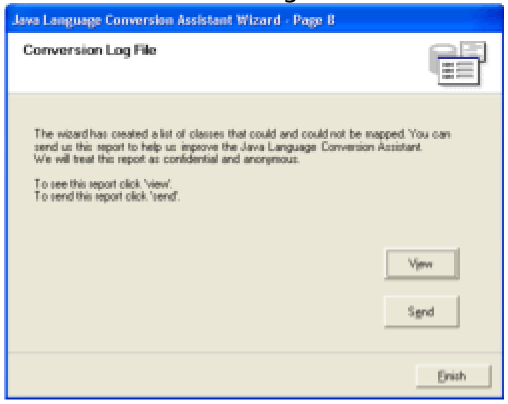
What name would you like for your .NET project?
MyProject.NET

What additional sources files are required by your project?
Enter the path to any additional source files required to build your project. Multiple paths may be separated by semi-colons.
C:\MyProject\MyProject.NET

What is the appropriate project type?
Windows Application

7. After the JLCA has collected all the information it needs, it will convert the project to a C# project in a new folder. By default, the wizard suggests creating a folder with the name <originalprojectname>.NET. Unless you really need to change it, accept the default folder name. Your old Java language project is left unchanged. The conversion process may take anywhere from a few minutes to several hours. During conversion, the progress bar will show you approximately how far through the process the conversion is.

8. After converting the project, the eighth and final page shows you that the conversion is finished, and gives you the opportunity to send a log file containing information about the conversion to Microsoft. By default, the log file is not automatically sent to Microsoft. Click the Send button to send it. For more about what is in the log file, see the section titled “Log File”

   The wizard has created a list of classes that could and could not be mapped. You can send us this report to help us improve the Java Language Conversion Assistant. We will treat this report as confidential and anonymous.

   To see this report click 'view'.
   To send this report click 'send'.

   Click finish to close the wizard

9. Your converted project will now open in the C# project explorer.

## Log File

The final page of the wizard allows you to send a log file about the conversion to Microsoft. So, what exactly is in this log file? Why should you send this information?

The JLCA generates a log file every time it does a conversion. This log file is always called “_ConversionSummary.txt”. The file contains the number and size of files in the project, the number of unrecognized classes, and the list of the JDK Level 1.1.4 classes used in the application. The file gives a summary of the composition of your project.

The file does NOT contain the name of the project, the name of any files, names of other non-JDK 1.1.4 classes, variables, or any other information about the project. It does not contain any code from your project.  There is no personal information in the file. It contains no identity information such as name, usercode or network user id. The following box shows the contents of a typical log file.

```
FileCount 2
Size 2138
Size 5256
com.ms.wfc.ui.Control.getText                1
com.ms.wfc.ui.Form.setAutoScaleBaseSize      1
com.ms.wfc.ui.Label                          3
com.ms.wfc.ui.Point.Point                    8
com.ms.wfc.core.Container.dispose            1
com.ms.wfc.ui.Form                           2
com.ms.wfc.ui.Edit.Edit                      1
com.ms.wfc.ui.Control.setSize                3
com.ms.wfc.ui.Point                          8
com.ms.wfc.ui.Control.setTabStop             1
java.lang.String                             2
```

The first section of the log file gives the number of files in the project and the sizes in bytes of each. This information is useful to ensure the JLCA is optimized for the right size of application. The final 11 lines show the JDK level 1.1.4 classes and methods used in the application, and the count for each. This information is useful to ensure the JLCA is optimized for the most commonly used classes. By sending this information to Microsoft, you are helping improve future releases of the JLCA.

### Sending the file yourself

If the wizard cannot send the log file to Microsoft, and you wish to send it yourself, please locate the file named _ConversionSummary.txt in the converted project, and email it to JLCAHelp@microsoft.com.

## Support

This is an unsupported Beta. Support for this beta is available through the MSNEWS newsgroup **microsoft.public.vsnet.jlca** only.  If you find bugs in the tool, you may submit them to JLCABug@microsoft.com. We cannot reply to every submission, but we will look at every submission.

## Limitations with This Release

The following issues all refer to features that are incomplete in Beta 2. Where noted, we will be adding support in a future release.

### JDK level 1.1.4

The JLCA is designed to convert many JDK level 1.1.4 classes. There are other commonly used classes outside of JDK level 1.1.4 with namespaces such as javax.*,org.omg.CORBA*, sun.*,sunw.* amd org.apache.*. The JLCA will not convert code that uses these classes. Statements that access these classes will be copied to C# unchanged. To finish the conversion, you will need to convert these statements to C# yourself.

### Project to project references

The JLCA is a single project converter. Project to project references are not resolved. If a project uses classes defined in another project, the statements that use the classes will not be converted; they will be copied to C# unchanged.

If a solution contains multiple projects, with project to project references, we recommend moving all the code, objects and files into a single project before converting. After converting to C#, the project can again be broken into multiple smaller projects to achieve the same structure as before.

### Libraries

Similar to project to project references, if your code accesses other libraries such as third party libraries or your own utilities, you have two options:

- If you have appropriate rights to convert the library source code, put the library source code into the project before conversion.
- If you don't have rights to convert the library source code, convert your project, and replace the statements that use the library with code that accessed the .NET framework

### Compiler Directives

The compiler directives #if, #elif, #else, and #endif are ignored during conversion. If a directive equates to false, the code it applies to is converted as though the directive itself equated to true. This can cause problems if a single code block spans an #if directive – some code may not be converted.  We recommend commenting out code in the false path of the compiler directive before converting. Support for handling compiler directives will be added in a future release.

## MTS and COM+ Transacted Components

This release does not fully convert MTS attributes and classes. The following three items need modification after the converter has finished:

1. GUIDs. The JLCA adds COM GUIDs to the converted classes and members. These should be removed
2. Assembly version.  The JLCA creates an Assembly version attribute of
   [assembly: AssemblyVersion("1.0.*")]
   This should be changed to
   [assembly: AssemblyVersion("1.0.0.0")]
   to prevent the version changing with each recompile
3. Change the client class creation code to use new <classname> instead of activation by ProgID

More support for automatically converting com.ms.mtx classes will be added in a future release.

## Convert Menu Item Missing

In some cases, after a successful installation, the JLCA may "disappear" from the Visual Studio .NET menu. This is caused by a registry issue in Visual Studio .NET.  If the JLCA disappears from the Visual Studio .NET menu, try uninstalling and re-installing the JLCA.  In some cases, the problem can be fixed by closing all instances of Visual Studio .NET, and running the following command from a command window:

```
DevEnv.Exe /Setup
```

## Additional Source Files Location and Application Type

For directory conversions, page four of the wizard prompts for the path to additional source files and also for the application type (Windows Application, Console Application, Class Library).  In Beta 2, these two options have no effect: the additional source files path is not used, and the converted application will always be a Windows EXE.  Support for these two options will be added in a future release.

## Help Links

This release includes more than 900 help topics; these are accessed from hyperlinks emitted into source code. Each issue is associated with a hyperlink to a help topic.  Please be aware that many topics are not yet available, but will be added in a future release.  In Beta2, the conversion report does not yet have these hyperlinks.